SQL PROJECT

Presenting

Project

On

Foundation of SQL

By :

Prajkta Radhakrishna Shinde

(Master's in Data Science)

Institute: ITVEDANT

Branch: Vashi

Email: Prashinde99@gmail.com

Professor.- Akansha Rane

19 november 2022

SQL PROJECT

Student Information System

Aim

Prime Aim of this project is to perform SQL Queries on different Organized Database with different kind of tables. And also my aim is to solve complex queries and sub-queries related to RDBMS.

Abstract:

Student information system helps us to get students details and course name , course fees are easily available for information purpose.This data is the most important thing in any organization and so it must be protected by malicious intended users. Apart from this course information tables use to check fees of any course.

SQL PROJECT

Introduction to SQL

SQL was earlier known as SEQUEL. SQL stands for structured query language. It is the language used to access the data and structures within a relational database. It is non-procedural.

It provides commands for the following tasks:

- Querying data
- Inserting, updating and deleting data
- Creating, modifying and deleting database objects
- Controlling access to the database and database objects.
- Guarantees database consistency.
- Monitoring database performance and configuration.

Structured Query Language is a computer language that we use to interact with a relational database. SQL is a tool for organizing, managing, and retrieving archived data from a computer database. The original name was given by IBM as Structured English Query Language, abbreviated by the acronym SEQUEL. When data needs to be retrieved from a database, SQL is used to make the request. The DBMS processes the SQL query retrieves the requested data and returns it to us. Rather, SQL statements describe how a collection of data should be organized or what data should be extracted or added to the database.

What is SQL?

- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

SQL PROJECT

SQL process:

- When an SQL command is executing for any RDBMS, then the system figure out the best way to carry out the request and the SQL engine determines that how to interpret the task.
- In the process, various components are included. These components can be optimization Engine, Query engine, Query dispatcher, classic, etc.
- All the non-SQL queries are handled by the classic query engine, but SQL query engine won't handle logical files.

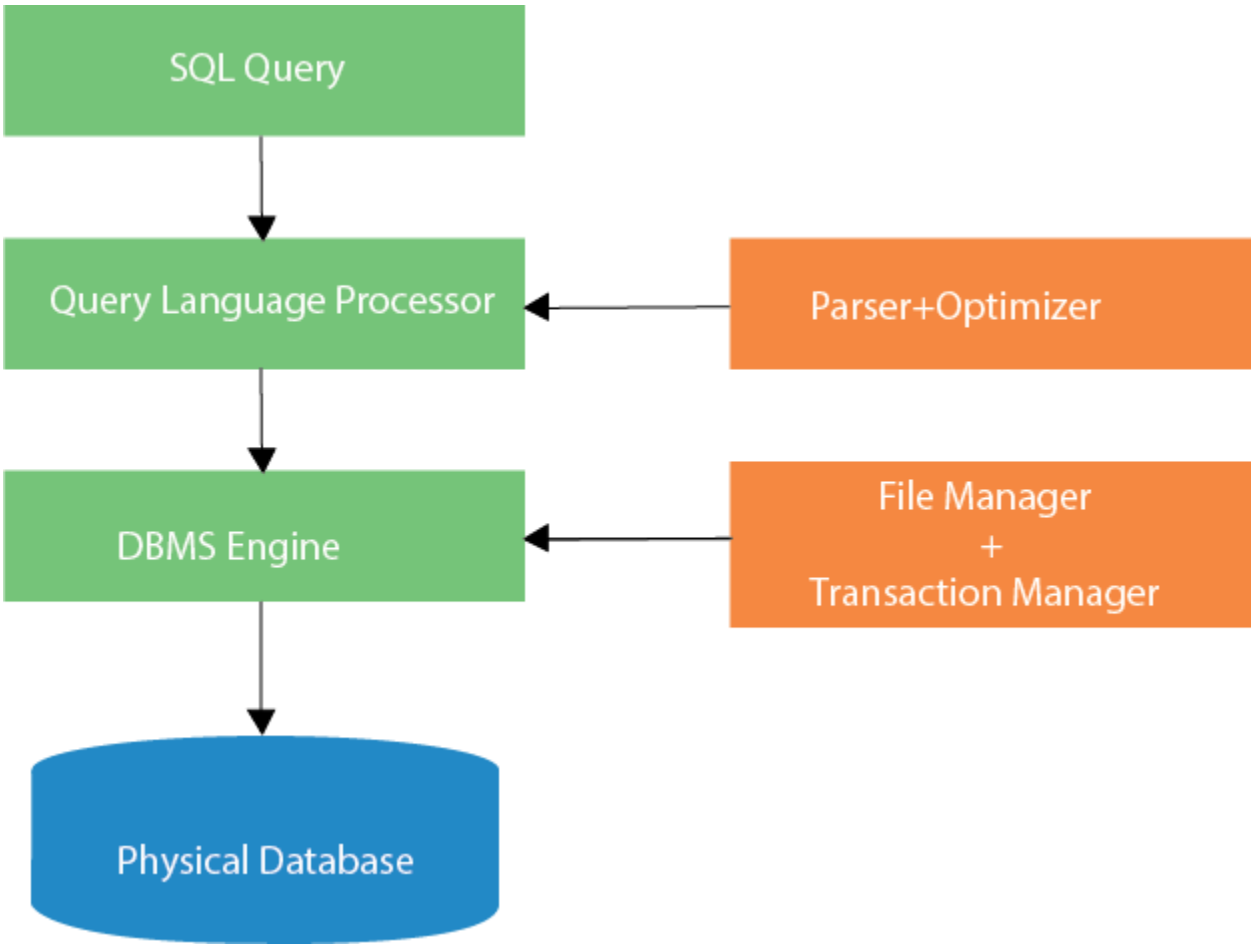

Fig. 1

SQL PROJECT

Types of SQL Commands:

1. DDL - Data Definition Language
   Description: Allows to create data structures i.e. tables, delete, alter etc.,
2. DQl - Data Query Language
   Description: Allows modification of data in the database
3. DML - Data Manipulation Language
   Description: Helps in Retrieving data and information from the database.
4. DCL - Data Control Language
   Description: Allows or denies permissions to access the tables.
5. TCL-Transaction Control Language (TCL)
   Description: Manages the changes made by the DML statements.

Types	Statements	Description
Data Definition Language (DDL)	1. Create 2. Alter 3. Drop 4. Truncate	This statement is used for defining the structure of a table.
Data Manipulation Language (DML)	1. Insert 2. Update 3. Delete	This statement is used to manipulate the data in the table.
Data Query Language (DQL)	1. Select 2. Show 3. Help	This stamen is for retrieve the data from the particular table according to the user need.
Data Control Language (DCL)	1. Grant 2. Revoke	Allows or denies the permission to access the data.
Transaction Control Language (TCL)	1. Commit 2. Rollback 3. Savepoint	Manages the changes made DML statements

Role of SQL :

SQL plays many different roles:

SQL PROJECT

• SQL is an interactive question language. Users type SQL instructions into an interactive SQL software to retrieve facts and show them on the screen, presenting a convenient, easy-to-use device for ad hoc database queries.
• SQL is a database programming language. Programmers embed SQL instructions into their utility packages to access the facts in a database. Both user-written packages and database software packages (consisting of document writers and facts access tools) use this approach for database access.
• SQL is a client/server language. Personal computer programs use SQL to communicate over a network with database servers that save shared facts. This client/server architecture is utilized by many famous enterprise-class applications.
• SQL is an Internet facts access language. Internet net servers that have interaction with company facts and Internet utility servers all use SQL as a widespread language for getting access to company databases, frequently through embedding SQL database get entry to inside famous scripting languages like PHP or Perl. pg. 4 | HIMANSHU KUMAR(LINKEDIN)
• SQL is a distributed database language. Distributed database control structures use SQL to assist distribute facts throughout many linked pc structures. The DBMS software program on every gadget makes use of SQL to speak with the opposite structures, sending requests for facts to get entry to.
• SQL is a database gateway language. In a pc community with a mixture of various DBMS products, SQL is frequently utilized in a gateway that lets in one logo of DBMS to speak with every other logo. SQL has for this reason emerged as a useful, effective device for linking people, pc packages, and pc structures to the facts saved in a relational database. Finally, SQL is not a particularly structured language, esp

Finally, SQL is not a particularly structured language, especially when compared with highly structured languages such as C, Pascal, or Java. Instead, SQL statements resemble English sentences, complete with “noise words” that don’t add to the meaning of the statement but make it read more naturally. The SQL has quite a few inconsistencies and also some special rules to prevent you from constructing SQL statements that look perfectly legal but that don’t make sense.

SQL uses:

• Data definition: It is used to define the structure and organization of the stored data and relationships among the stored data items.
• Data retrieval: SQL can also be used for data retrieval.
• Data manipulation: If the user wants to add new data, remove data, or modifying in existing data then SQL provides this facility also.
• Access control: SQL can be used to restrict a user’s ability to retrieve, add, and modify data, protecting stored data against unauthorized access.
• Data sharing: SQL is used to coordinate data sharing by concurrent users, ensuring that changes made by one user do not inadvertently wipe out changes made at nearly the same time by another user. SQL also differs from other computer languages because it describes what the user wants the computer to do rather than how the computer should do it. (In more technical terms, SQL is a declarative or descriptive language rather than a procedural one.) SQL contains no IF statement for testing conditions, and no GOTO, DO, or FOR statements for program flow control. Rather, SQL statements describe how a collection of data is to be organized, or what data is to be retrieved or added to the database. The sequence of steps to do those tasks is left for the DBMS to determine.

SQL PROJECT

What is Data?

Data is a collection of a distinct small unit of information. It can be used in a variety of forms like text, numbers, media, bytes, etc. it can be stored in pieces of paper or electronic memory, etc.

Word 'Data' is originated from the word 'datum' that means 'single piece of information.' It is plural of the word datum.

In computing, Data is information that can be translated into a form for efficient movement and processing. Data is interchangeable.

What is Database?

A **database** is an organized collection of data, so that it can be easily accessed and managed

Flow of Sql Queries

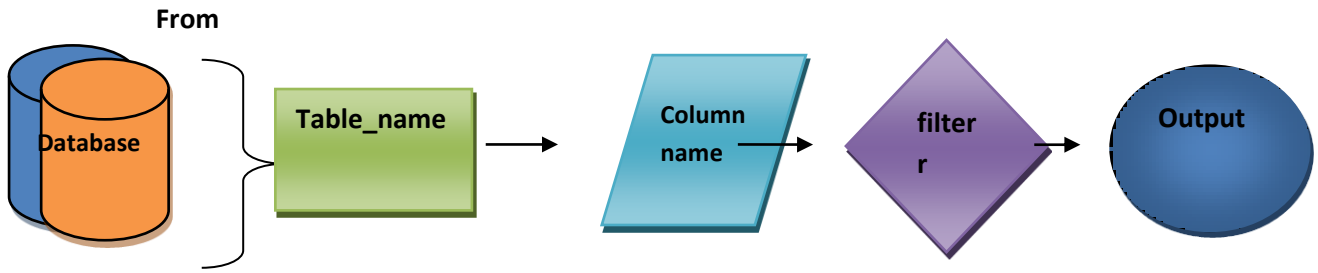

Fig. 2

SQL PROJECT

Studentsinfo table

Std_id	Std_name	Std_address	Age
1	Harsh	Delhi	18
25	Pratik	Bihar	20
26	Riyanaka	siliguri	24
27	Harshika	Ramnagar	19
28	Rohit	Ramnagar	19
29	Niraj	balurghat	23
30	Omkar	nashik	23
31	kimaya	Mumbai	25
32	Hruta	Pune	22
33	Ramesh	Punjab	21
34	Reva	Nashik	17

COURSE_INFO Table

C_id	C_name	C_Fees	Std_id
11	Python	10000	25
12	DBMS	5000	26
13	Data science	700000	27
14	SQL	4500	28
15	JAVA	5800	29
16	C++	4755	30
17	Machine learing	9877	31
18	It	5890	32

SQL PROJECT

CREATE DATABASE

```
Create database COURSE_DETAILS
```

CREATE TABLE

1.STUDENTSINFO

```
Create table studentsinfo(std_id int not null  primary key, std_name
varchar(25), std_address varchar(25), age int);
```

2.COURSE_INFO

```
Create table course_info(c_id int not null primary key, c_NAME
varchar(25), c_fees int, std_id int, foregine key(std_int) reference
studentsinfo(std_id));
```

QUERIES

1.Retrive entire contents form Studentsinfo

```
Syntax: select * from studentsinfo;
```

SQL PROJECT

std_id	std_name	std_address	age
1	harsh	delhi	18
25	pratik	bihar	20
26	riyanka	siliguri	24
27	harshika	ramnagar	19
28	rohit	ramnagar	19
29	niraj	balurghat	23
30	omkar	nashik	23
31	kimaya	mumbai	25
32	hruta	pune	22
33	ramesh	punjab	21
34	reva	nashik	17

2. Find out the names of all the students

Syntax: `select std_name from studentsinfo;`

```
MariaDB [course_details]> select std_name from studentsinfo;
+----------+
| std_name |
+----------+
| harsh    |
| pratik   |
| riyanka  |
| harshika |
| rohit    |
| niraj    |
| omkar    |
| kimaya   |
| hruta    |
| ramesh   |
| reva     |
+----------+
11 rows in set (0.001 sec)
```

SQL PROJECT

3.Retrive the list of studentid and age from the studentsinfo

Synatx: select std id, age from studentsinfo;

```
MariaDB [course_details]> select std_id, age from studentsinfo;
+--------+-----+
| std_id | age |
+--------+-----+
|      1 |  18 |
|     25 |  20 |
|     26 |  24 |
|     27 |  19 |
|     28 |  19 |
|     29 |  23 |
|     30 |  23 |
|     31 |  25 |
|     32 |  22 |
|     33 |  21 |
|     34 |  17 |
+--------+-----+
11 rows in set (0.001 sec)
```

4.Retrive entire contents form Studentsinfo

Synatx: select * from course_info;

```
MariaDB [course_details]> select * from course_info
    -> ;
+------+-----------------+--------+--------+
| c_id | c_name          | c_fees | std_id |
+------+-----------------+--------+--------+
|   11 | python          |  10000 |     25 |
|   12 | DBMS            |   5000 |     26 |
|   13 | data science    | 700000 |     27 |
|   14 | SQL             |   4500 |     28 |
|   15 | JAVA            |   5800 |     29 |
|   16 | C++             |   4755 |     30 |
|   17 | machine learing |   9877 |     31 |
|   18 | it              |   5890 |     32 |
+------+-----------------+--------+--------+
8 rows in set (0.000 sec)
```

5. find out the name of all the course.

Syntax: select c_name from course_info;

SQL PROJECT

```
MariaDB [course_details]> select c_name from course_info
    -> ;
+-----------------+
| c_name          |
+-----------------+
| python          |
| DBMS            |
| data science    |
| SQL             |
| JAVA            |
| C++             |
| machine learing |
| it              |
+-----------------+
8 rows in set (0.000 sec)
```

6.student from studentsinfo who are located in mumbai

Syntax: select std_name from studentsinfo where std_address="mumbai";

```
MariaDB [course_details]> select std_name from studentsinfo where std_address="mumbai";
+----------+
| std_name |
+----------+
| kimaya   |
+----------+
1 row in set (0.001 sec)
```

7. create table contact no in studentsinfo

Syntax: alter table studentsinfo add contactno bigint;

SQL PROJECT

```
MariaDB [course_details]> alter table studentsinfo add contactno bigint;
Query OK, 0 rows affected (0.053 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [course_details]> select * from studentsinfo;
+--------+----------+-------------+-----+-----------+
| std_id | std_name | std_address | age | contactno |
+--------+----------+-------------+-----+-----------+
|      1 | harsh    | delhi       |  18 |      NULL |
|     25 | pratik   | bihar       |  20 |      NULL |
|     26 | riyanka  | siliguri    |  24 |      NULL |
|     27 | harshika | ramnagar    |  19 |      NULL |
|     28 | rohit    | ramnagar    |  19 |      NULL |
|     29 | niraj    | balurghat   |  23 |      NULL |
|     30 | omkar    | raj         |  23 |      NULL |
|     31 | kimaya   | mumbai      |  25 |      NULL |
|     32 | hruta    | pune        |  22 |      NULL |
|     33 | ramesh   | punjab      |  21 |      NULL |
|     34 | reva     | nashik      |  17 |      NULL |
+--------+----------+-------------+-----+-----------+
11 rows in set (0.001 sec)
```

8.insert contact no in studentsinfo.

Syntax: update studentsinfo set contactno=(case when std_id=1 then 8954234896 when std_id=25 then 8954567896 when std_id=26 then 8954234896 when std_id=27 then 8954232156 when std_id=28 then 7896142351 when std_id=29 then 7456981236 when std_id=30 then 4569871236 when std_id=31 then 7456289137 when std_id=32 then 4796321567 when std_id=33 then 8569741325 when std_id=34 then 6985741236 END) where std_id in(1,25,26,27,28,29,30,31,32,33,34);

SQL PROJECT

```
MariaDB [course_details]> select * from studentsinfo;
+--------+----------+-------------+-----+------------+
| std_id | std_name | std_address | age | contactno  |
+--------+----------+-------------+-----+------------+
|      1 | harsh    | delhi       |  18 | 8954234896 |
|     25 | pratik   | bihar       |  20 | 8954567896 |
|     26 | riyanka  | siliguri    |  24 | 8954234896 |
|     27 | harshika | ramnagar    |  19 | 8954232156 |
|     28 | rohit    | ramnagar    |  19 | 7896142351 |
|     29 | niraj    | balurghat   |  23 | 7456981236 |
|     30 | omkar    | nashik      |  23 | 4569871236 |
|     31 | kimaya   | mumbai      |  25 | 7456289137 |
|     32 | hruta    | pune        |  22 | 4796321567 |
|     33 | ramesh   | punjab      |  21 | 8569741325 |
|     34 | reva     | nashik      |  17 | 6985741236 |
+--------+----------+-------------+-----+------------+
11 rows in set (0.001 sec)
```

9.DISTINCT

Syntax: select distinct std_address from studentsinfo;

```
MariaDB [course_details]> select distinct std_address from studentsinfo;
+-------------+
| std_address |
+-------------+
| delhi       |
| bihar       |
| siliguri    |
| ramnagar    |
| balurghat   |
| nashik      |
| mumbai      |
| pune        |
| punjab      |
+-------------+
9 rows in set (0.001 sec)
```

10. IN

Syntax: select std_name from studentsinfo where std_name in("hruta", "rohit", "harsh");

SQL PROJECT

```
MariaDB [course_details]> select std_name from studentsinfo where std_name in("hruta", "rohit", "harsh");
+----------+
| std_name |
+----------+
| harsh    |
| rohit    |
| hruta    |
+----------+
3 rows in set (0.001 sec)
```

11. GROUP BY

Syntax: select std_id, count(std_address) from studentsinfo group by std_address;

```
MariaDB [course_details]> select std_id, count(std_address) from studentsinfo group by std_address;
+--------+--------------------+
| std_id | count(std_address) |
+--------+--------------------+
|     29 |                  1 |
|     25 |                  1 |
|      1 |                  1 |
|     31 |                  1 |
|     30 |                  2 |
|     32 |                  1 |
|     33 |                  1 |
|     27 |                  2 |
|     26 |                  1 |
+--------+--------------------+
9 rows in set (0.001 sec)
```

12. INNER JOIN

Syntax: select studentsinfo.std_id, studentsinfo.std_name, studentsinfo.contactno, course_info.c_fees from studentsinfo INNER JOIN course_info on studentsinfo.std_id=course_info.std_id;

```
MariaDB [course_details]> select studentsinfo.std_id,  studentsinfo.std_name, studentsinfo.contactno, course_info.c_fees from studentsinfo INNER JOIN course_info on studentsinfo.std_id=course_info.std_id;
+--------+----------+------------+--------+
| std_id | std_name | contactno  | c_fees |
+--------+----------+------------+--------+
|     25 | pratik   | 8954567896 |  10000 |
|     26 | riyanka  | 8954234896 |   5000 |
|     27 | harshika | 8954232156 | 700000 |
|     28 | rohit    | 7896142351 |   4500 |
|     29 | niraj    | 7456981236 |   5800 |
|     30 | omkar    | 4569871236 |   4755 |
|     31 | kimaya   | 7456289137 |   9877 |
|     32 | hruta    | 4796321567 |   5890 |
+--------+----------+------------+--------+
8 rows in set (0.001 sec)
```

13. LEFT OUTER JOIN

Syntax: select studentsinfo.std_id, studentsinfo.std_name, studentsinfo.age, course_info.c_id, course_info.c_name from studentsinfo left outer join course_info on studentsinfo.std_id=course_info.std_id;

SQL PROJECT

```
MariaDB [course_details]>  select studentsinfo.std_id, studentsinfo.std_name, studentsinfo.age, course_info.c_id, course_info.c_name from studentsinfo left outer join course_info on studentsinfo.std_id=course_in
fo.std_id;
+--------+----------+-----+------+----------------+
| std_id | std_name | age | c_id | c_name         |
+--------+----------+-----+------+----------------+
|      1 | harsh    |  18 | NULL | NULL           |
|     25 | pratik   |  20 |   11 | python         |
|     26 | riyanka  |  24 |   12 | DBMS           |
|     27 | harshika |  19 |   13 | data science   |
|     28 | rohit    |  19 |   14 | SQL            |
|     29 | niraj    |  23 |   15 | JAVA           |
|     30 | omkar    |  23 |   16 | C++            |
|     31 | kimaya   |  25 |   17 | machine learing |
|     32 | hruta    |  22 |   18 | it             |
|     33 | ramesh   |  21 | NULL | NULL           |
|     34 | reva     |  17 | NULL | NULL           |
+--------+----------+-----+------+----------------+
11 rows in set (0.002 sec)
```

14. RIGHT OUTER JOIN

Syntax: select studentsinfo.std_id, studentsinfo.std_name, studentsinfo.age, course_info.c_id, course_info.c_name from studentsinfo right outer join course_info on studentsinfo.std_id=course_info.std_id;

```
MariaDB [course_details]> select studentsinfo.std_id, studentsinfo.std_name, studentsinfo.age, course_info.c_id, course_info.c_name from studentsinfo right outer join course_info on studentsinfo.std_id=course_in
fo.std_id;
+--------+----------+-----+------+----------------+
| std_id | std_name | age | c_id | c_name         |
+--------+----------+-----+------+----------------+
|     25 | pratik   |  20 |   11 | python         |
|     26 | riyanka  |  24 |   12 | DBMS           |
|     27 | harshika |  19 |   13 | data science   |
|     28 | rohit    |  19 |   14 | SQL            |
|     29 | niraj    |  23 |   15 | JAVA           |
|     30 | omkar    |  23 |   16 | C++            |
|     31 | kimaya   |  25 |   17 | machine learing |
|     32 | hruta    |  22 |   18 | it             |
+--------+----------+-----+------+----------------+
8 rows in set (0.037 sec)
```

16. Add column marks

Syntax: alter table studentsinfo add marks int;

```
MariaDB [course_details]> select * from studentsinfo;
+--------+----------+-------------+-----+------------+--------+
| std_id | std_name | std_address | age | contactno  | marks  |
+--------+----------+-------------+-----+------------+--------+
|      1 | harsh    | delhi       |  18 | 8954234896 |  NULL  |
|     25 | pratik   | bihar       |  20 | 8954567896 |  NULL  |
|     26 | riyanka  | siliguri    |  24 | 8954234896 |  NULL  |
|     27 | harshika | ramnagar    |  19 | 8954232156 |  NULL  |
|     28 | rohit    | ramnagar    |  19 | 7896142351 |  NULL  |
|     29 | niraj    | balurghat   |  23 | 7456981236 |  NULL  |
|     30 | omkar    | nashik      |  23 | 4569871236 |  NULL  |
|     31 | kimaya   | mumbai      |  25 | 7456289137 |  NULL  |
|     32 | hruta    | pune        |  22 | 4796321567 |  NULL  |
|     33 | ramesh   | punjab      |  21 | 8569741325 |  NULL  |
|     34 | reva     | nashik      |  17 | 6985741236 |  NULL  |
+--------+----------+-------------+-----+------------+--------+
11 rows in set (0.000 sec)
```

17. To adding the data into marks column.

Syntax: update studentsinfo set marks=(case when std_id=1 then 45 when std_id=25 then 78 when std_id=26 then 48 when std_id=27 then 45 when std_id=28 then 96 when std_id=29 then 85 when

SQL PROJECT

std_id=30 then 74 when std_id=31 then 62 when std_id=32 then 56 when std_id=33 then 76 when std_id=34 then 85 END)where std_id in(1,25,26,27,28,29,30,31,32,33,34);

```
MariaDB [course_details]> select * from studentsinfo;
+--------+----------+-------------+-----+------------+-------+
| std_id | std_name | std_address | age | contactno  | marks |
+--------+----------+-------------+-----+------------+-------+
|      1 | harsh    | delhi       |  18 | 8954234896 |    45 |
|     25 | pratik   | bihar       |  20 | 8954567896 |    78 |
|     26 | riyanka  | siliguri    |  24 | 8954234896 |    48 |
|     27 | harshika | ramnagar    |  19 | 8954232156 |    45 |
|     28 | rohit    | ramnagar    |  19 | 7896142351 |    96 |
|     29 | niraj    | balurghat   |  23 | 7456981236 |    85 |
|     30 | omkar    | nashik      |  23 | 4569871236 |    74 |
|     31 | kimaya   | mumbai      |  25 | 7456289137 |    62 |
|     32 | hruta    | pune        |  22 | 4796321567 |    56 |
|     33 | ramesh   | punjab      |  21 | 8569741325 |    76 |
|     34 | reva     | nashik      |  17 | 6985741236 |    85 |
+--------+----------+-------------+-----+------------+-------+
11 rows in set (0.000 sec)
```

18. Order by asc

Syntax: select * from studnetsinfo order by std_name ASC;

```
MariaDB [course_details]> select * from studentsinfo order by std_name ASC;
+--------+----------+-------------+-----+------------+-------+
| std_id | std_name | std_address | age | contactno  | marks |
+--------+----------+-------------+-----+------------+-------+
|      1 | harsh    | delhi       |  18 | 8954234896 |    45 |
|     27 | harshika | ramnagar    |  19 | 8954232156 |    45 |
|     32 | hruta    | pune        |  22 | 4796321567 |    56 |
|     31 | kimaya   | mumbai      |  25 | 7456289137 |    62 |
|     29 | niraj    | balurghat   |  23 | 7456981236 |    85 |
|     30 | omkar    | nashik      |  23 | 4569871236 |    74 |
|     25 | pratik   | bihar       |  20 | 8954567896 |    78 |
|     33 | ramesh   | punjab      |  21 | 8569741325 |    76 |
|     34 | reva     | nashik      |  17 | 6985741236 |    85 |
|     26 | riyanka  | siliguri    |  24 | 8954234896 |    48 |
|     28 | rohit    | ramnagar    |  19 | 7896142351 |    96 |
+--------+----------+-------------+-----+------------+-------+
11 rows in set (0.001 sec)
```

19. ORDER BY DESC

Syntax: select * from studentsinfo order by std_name DESC;

SQL PROJECT

```
MariaDB [course_details]> select * from studentsinfo order by std_name DESC;
+--------+----------+-------------+-----+------------+-------+
| std_id | std_name | std_address | age | contactno  | marks |
+--------+----------+-------------+-----+------------+-------+
|     28 | rohit    | ramnagar    |  19 | 7896142351 |    96 |
|     26 | riyanka  | siliguri    |  24 | 8954234896 |    48 |
|     34 | reva     | nashik      |  17 | 6985741236 |    85 |
|     33 | ramesh   | punjab      |  21 | 8569741325 |    76 |
|     25 | pratik   | bihar       |  20 | 8954567896 |    78 |
|     30 | omkar    | nashik      |  23 | 4569871236 |    74 |
|     29 | niraj    | balurghat   |  23 | 7456981236 |    85 |
|     31 | kimaya   | mumbai      |  25 | 7456289137 |    62 |
|     32 | hruta    | pune        |  22 | 4796321567 |    56 |
|     27 | harshika | ramnagar    |  19 | 8954232156 |    45 |
|      1 | harsh    | delhi       |  18 | 8954234896 |    45 |
+--------+----------+-------------+-----+------------+-------+
11 rows in set (0.000 sec)
```

20. count

Syntax: SELECT count(*) from studentsinfo;

```
MariaDB [course_details]> SELECT count(*) from studentsinfo;
+----------+
| count(*) |
+----------+
|       11 |
+----------+
1 row in set (0.001 sec)
```

21. Sum

Syntax: select sum(marks) from studentsinfo;

```
MariaDB [course_details]> select sum(marks) from studentsinfo;
+------------+
| sum(marks) |
+------------+
|        750 |
+------------+
1 row in set (0.000 sec)
```

22. Average

Syntax: select avg(marks) from studentsinfo;

SQL PROJECT

```
MariaDB [course_details]> select avg(marks) from studentsinfo;
+------------+
| avg(marks) |
+------------+
|    68.1818 |
+------------+
1 row in set (0.000 sec)
```

23. min

Syntax: select min(marks) from studentsinfo;

```
MariaDB [course_details]>  select min(marks) from studentsinfo;
+------------+
| min(marks) |
+------------+
|         45 |
+------------+
1 row in set (0.001 sec)
```

24.max

Syntax: select max(marks) from studentsinfo;

```
MariaDB [course_details]>  select max(marks) from studentsinfo;
+------------+
| max(marks) |
+------------+
|         96 |
+------------+
1 row in set (0.000 sec)
```

25. Between

Syntax: select * from studentsinfo where age between 20 and 23;

```
MariaDB [course_details]> select * from studentsinfo where age between 20 and 23;
+--------+----------+-------------+-----+------------+-------+
| std_id | std_name | std_address | age | contactno  | marks |
+--------+----------+-------------+-----+------------+-------+
|     25 | pratik   | bihar       |  20 | 8954567896 |    78 |
|     29 | niraj    | balurghat   |  23 | 7456981236 |    85 |
|     30 | omkar    | nashik      |  23 | 4569871236 |    74 |
|     32 | hruta    | pune        |  22 | 4796321567 |    56 |
|     33 | ramesh   | punjab      |  21 | 8569741325 |    76 |
+--------+----------+-------------+-----+------------+-------+
5 rows in set (0.001 sec)
```

SQL PROJECT

26. view- single table

Syntax: create view students as select std_name, std_id, marks from studentsinfo where marks <=56;

```
MariaDB [course_details]> select *from students;
+----------+--------+-------+
| std_name | std_id | marks |
+----------+--------+-------+
| harsh    |      1 |    45 |
| riyanka  |     26 |    48 |
| harshika |     27 |    45 |
| hruta    |     32 |    56 |
+----------+--------+-------+
4 rows in set (0.001 sec)
```

27. view-multiple table

Syntax: create view detailsview as select studentsinfo.std_name, studentsinfo.marks, course_info.c_fees, course_info.c_name from studentsinfo, course_info where studentsinfo.std_id=course_info.std_id;

```
MariaDB [course_details]> select * from detailsview;
+----------+-------+--------+-----------------+
| std_name | marks | c_fees | c_name          |
+----------+-------+--------+-----------------+
| pratik   |    78 |  10000 | python          |
| riyanka  |    48 |   5000 | DBMS            |
| harshika |    45 | 700000 | data science    |
| rohit    |    96 |   4500 | SQL             |
| niraj    |    85 |   5800 | JAVA            |
| omkar    |    74 |   4755 | C++             |
| kimaya   |    62 |   9877 | machine learing |
| hruta    |    56 |   5890 | it              |
+----------+-------+--------+-----------------+
8 rows in set (0.001 sec)
```

28. drop view

Syntax: drop view detailsview;

```
MariaDB [course_details]> drop view detailsview;
Query OK, 0 rows affected (0.001 sec)

MariaDB [course_details]> select *  from detailsview;
ERROR 1146 (42S02): Table 'course_details.detailsview' doesn't exist
```

29. `select std_name, marks, std_id from studentsinfo where age<=18;`

SQL PROJECT

std_name	marks	std_id
harsh	45	1
reva	85	34

Conclusion

I hereby conclude that I have completed my project and achieved my aim by performing all the SQL queries to best of my knowledge**.**